MITIGATING POLICY-REWARD MISMATCH IN SKILL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised skill discovery (SD) aims to enable agents to explore in environments without task rewards, where a key factor determining their utility on downstream tasks is the joint state space coverage of all learned skills. However, we find that even state-of-the-art methods cannot sufficiently cover the state space. Gaps between skills formed in the early learning stage are never filled, which significantly reduces the utility of skills on downstream tasks. In this paper, we provide an in-depth theoretical analysis of this *dead-zone* phenomenon. We identify that a core issue is **policy-reward mismatch**: the intrinsic reward mismatches the policy's exploration needs. When the policy initially explores a new region, the skill discriminator is dominated by early data and fails to update the reward for the new region in time, which suppresses further exploration. To address this issue, we propose RASD, a simple yet efficient method that provides more on-policy data for the policy and skill discriminator to ensure quick reward adaptation. Extensive experiments show that our method effectively improves the state-space coverage of skills and achieves better performance on downstream tasks.

1 INTRODUCTION

Reinforcement learning has demonstrated strong performance in many domains (Silver et al., 2016). However, the success depends on carefully designed reward functions, which require substantial manual effort. An effective solution to this issue is to use unsupervised training as a pretraining stage (Brown et al., 2020; Xie et al., 2022). As a representative unsupervised reinforcement learning method, skill discovery (SD) aims to enable agents to explore in environments without task rewards (Eysenbach et al., 2019; Laskin et al., 2022), learn task-agnostic knowledge, and form diverse and distinguishable behaviors as skills. These skills can then be reused directly or leveraged to efficiently train a high-level policy on top of them for downstream tasks with specific task rewards.

A key factor determining the utility of unsupervised skill discovery methods on downstream tasks is the joint state space coverage of all learned skills (Eysenbach et al., 2019; Strouse et al., 2022). Early mutual-information-based skill discovery can learn only static behaviors in complex environments, prompting subsequent improvements (Laskin et al., 2022; Park et al., 2022). However, we find that even state-of-the-art methods cannot sufficiently cover the state space. A typical phenomenon is that although skills continually improve their coverage during training, gaps between skills formed in the early learning stage are never filled, as shown in Fig. 1. We term this the *dead zone* phenomenon. This significantly reduces the utility of skills on downstream tasks.

In this paper, we provide an in-depth theoretical analysis of this problem. Building on METRA (Park et al., 2024), we characterize the evolving behavior of the intrinsic reward and the policy during skill discovery. Our analysis shows that a key mechanism causing the dead zone is **policy-reward mismatch**: the intrinsic reward mismatches the policy's exploration needs. It fails to encourage policy exploration (Prop. 2). Instead, optimizing this reward traps the policy in a local optimum and suppresses further exploration (Prop. 3). As the behavior policy gradually approaches the local optima on the collected data, its exploration ability weakens, and thus it is no longer able to collect enough new data to escape local optima (Prop. 4).

The policy–reward mismatch arises because the skill discriminator is overwhelmed by a large volume of already-mastered early data (Prop. 1), thereby failing to update rewards for the exploration frontier in a timely manner. To address this issue, we propose **Reward-Aligned Skill Discovery**

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

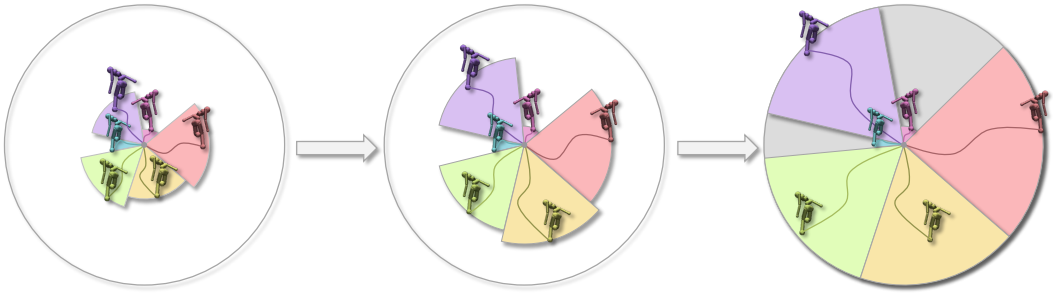


Figure 1: Illustration of the dead zone phenomenon. The disk represents the state space, and each color denotes a skill. Each skill (color) aims to cover the sector of the same color, with all skills jointly covering the entire disk. During the intermediate stage of skill discovery, some skills stop expanding within the state space, leaving the gray areas in the right subfigure uncovered by any skill. We term these gray areas dead zones.

(RASD), which resolves the mismatch by emphasizing on-policy data. By providing more on-policy data for policy and reward updates, it mitigates exploration suppression caused by delayed reward updates when the policy attempts to explore new regions, thereby further improving the policy's exploration ability in the early epochs. Experiments on DMControl and MuJoCo tasks demonstrate that RASD can improve skill coverage more rapidly, mitigate the dead-zone phenomenon, and achieve better final performance.

In summary, our contributions are threefold:

- We identify the policy-reward mismatch problem in skill discovery and, through a thorough theoretical analysis, show that it is a key reason for the dead-zone phenomenon.
- We propose RASD, a simple yet efficient method for addressing the aforementioned issue. RASD provides more on-policy data for the policy and the skill encoder to ensure quick intrinsic reward adaptation during exploration.
- Extensive experiments show that our method effectively improves the state-space coverage of skills compared with existing skill discovery methods, mitigates the dead-zone phenomenon, and achieves better performance on downstream tasks.

2 Preliminaries

2.1 Unsupervised Skill Discovery

Skill discovery (SD) is an unsupervised reinforcement learning (URL) method that builds on the reward-free Markov decision process (MDP). Here, the MDP is characterized by state space $\mathcal{S}$, action space $\mathcal{A}$, state transition $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and initial state distribution μ_0 (Xie et al., 2022). The target of SD is to learn multiple diverse and distinguishable policies as skills to explore the state space, which can then be directly reused in a zero-shot manner or used as lower-level primitives in hierarchical policies, facilitating downstream tasks.

To achieve this, SD methods learn a skill-conditioned policy $\pi(a|s, z)$, where z denotes the skill latent, and construct intrinsic rewards with empowerment measures to encourage exploration and ensure the distinguishability of learned skills. Empowerment measures the degree to which the policy can control the environment (Gregor et al., 2016). A typical empowerment measure is the mutual information (MI) between skills z and the states along the trajectory, denoted as $I(s, z)$, which can be optimized using a skill discriminator $q(z|s)$ or $q(z|s, s')$ (Eysenbach et al., 2019; Zheng et al., 2025), or a state density model $q(s|z)$ (Liu & Abbeel, 2021; Laskin et al., 2022).

2.2 Distance-maximizing Skill Discovery

Distance-maximizing skill discovery (DSD) methods have recently been proposed to address the static-behavior issue in MI-based skill discovery methods (Park et al., 2024). DSD methods en-

courage coverage in state space by linking the distance in latent space with that in state space. The objective of DSD methods is shown as follows:

$$\sup_{\pi,\phi} \mathbb{E}_{\tau,z}\left[\sum_{t=0}^{T-1} \left(\phi(s_{t+1}) - \phi(s_t)\right)^\top z\right] \quad \text{s.t. } \|\phi(x) - \phi(y)\|_2 \le d(x,y),\ \forall x, y \in \mathcal{S}, \tag{1}$$

where τ is the state trajectory, $d(\cdot,\cdot)$ is an arbitrary non-negative function serving as a distance metric. The intrinsic reward is $r(s, s', z) = (\phi(s') - \phi(s))^\top z$, $\|z\|_2 = 1$.

A representative DSD method, METRA (Park et al., 2024), uses temporal distance as the distance metric in Eq. 1. The full objective is as follows, which can be optimized by introducing a Lagrange multiplier λ and adopting dual gradient descent.

$$\sup_{\pi,\phi} \mathbb{E}_{\tau,z}\left[\sum_{t=0}^{T-1} \left(\phi(s_{t+1}) - \phi(s_t)\right)^\top z\right] \quad \text{s.t. } \|\phi(s') - \phi(s)\|_2 \le 1,\ \forall (s, s') \in \mathcal{S}_{\text{adj}}, \tag{2}$$

$$\mathcal{J}^\phi = \mathbb{E}_{(s,z,s')\sim\mathcal{D}}[(\phi(s') - \phi(s))^\top z + \lambda \min(\epsilon, 1 - \|\phi(s') - \phi(s)\|_2^2)], \tag{3}$$

$$\mathcal{J}^\lambda = \mathbb{E}_{(s,z,s')\sim\mathcal{D}}[\lambda \min(\epsilon, 1 - \|\phi(s') - \phi(s)\|_2^2)], \tag{4}$$

where $\mathcal{D}$ is the replay buffer and $\mathcal{S}_{\text{adj}}$ denotes the set of adjacent state pairs within trajectories.

METRA is shown to be largely explained in the MI framework in Zheng et al. (2025). Specifically, Zheng et al. (2025) shows that the encoder training of METRA can be approximated by constructing the skill discriminator $q(z|s, s')$ using a scaled energy-based model and METRA's intrinsic reward

$$q_\phi(z \mid s, s') \triangleq \frac{p(z)\exp((\phi(s') - \phi(s))^\top z)}{\mathbb{E}_{p(z')}[\exp((\phi(s') - \phi(s))^\top z')]}, \tag{5}$$

and then updating it using a contrastive learning algorithm. Additionally, the policy's objective function in METRA can be approximated by an information bottleneck, which removes the anti-exploration term from the mutual information between the skill latent and the transition.

3 Policy-Reward Mismatch

In this section, we conduct theoretical analysis to characterize the behaviors of the skill discovery algorithm leading to the dead zone phenomenon. We identify the core issue as **policy-reward mismatch**: the intrinsic reward mismatches the policy's exploration needs. Specifically, the intrinsic reward fails to encourage exploration (Proposition 2), instead trapping the policy in a local optimum and suppressing further exploration (Proposition 3). As the behavior policy gradually approaches the local optima on the collected data, its exploration ability weakens, preventing it from collecting enough new data to escape that optimum (Proposition 4).

A straightforward reason for the dead zone phenomenon is that early data are of low quality and limited coverage, so the encoder and policy can only learn suboptimal behavior. Therefore, we assume transitions collected in early epochs are restricted to a state subset denoted by $\mathcal{S}_{\text{early}}$, with the uncovered state set defined as $\mathcal{S}_{\text{late}} = \mathcal{S} \setminus \mathcal{S}_{\text{early}}$. Proposition 1 provides further evidence from the perspective of how data affect gradient updates.

Proposition 1. *Consider a replay buffer $\mathcal{D}$ storing N transitions. At each environment step, k gradient updates are performed, and each gradient update uses a mini-batch of size b sampled uniformly from $\mathcal{D}$. Let $\beta = bk$. For the t-th transition added to the buffer $(1 < t \le N)$, the expected number of times it is sampled during training, $\mathbb{E}[n_t]$, satisfies*

$$\mathbb{E}[n_t] = \Theta\left(\log \frac{N}{t}\right). \tag{6}$$

Proposition 1 indicates that throughout the training process, early collected transitions (with small t) are sampled much more frequently than later collected transitions. Since sampling frequency corresponds in expectation to the weight of each data point in gradient updates, the higher sampling frequency of early data implies that gradients from early data are weighted more heavily throughout training, which means that the gradient updates of the skill encoder ϕ are dominated by early data.

Next, we analyze the intrinsic reward induced by the restricted early-epoch data in Proposition 2.

Proposition 2. *For any transition pair (s, s'), if the variational posterior satisfies $D_{\mathrm{KL}}(q_\phi(z \mid s, s') \| p(z)) \le \epsilon_{s,s'}$, where $\epsilon_{s,s'} \ge 0$, $p(z) = \mathrm{UNIF}(\mathbb{S}^{d-1})$ (d is the dimension of the skill latent z), then the norm of the representation difference satisfies*

$$\|\phi(s') - \phi(s)\|_2 \le \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}), \tag{7}$$

and the intrinsic reward satisfies

$$\sup_{z \in \mathbb{S}^{d-1}} |r(s, s', z)| \le \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}). \tag{8}$$

In particular, if $\epsilon_{s,s'} = 0$, then the posterior cannot deviate from the prior, so $\phi(s') = \phi(s)$, and for all $z \in \mathbb{S}^{d-1}$ we have $r(s, s', z) = 0$.

Proposition 2 indicates that once the encoder is trained, the intrinsic reward derived from the encoder is upper bounded by the KL divergence between the prior and posterior of the discriminator's output distribution. As early data (in $\mathcal{S}_{\text{early}}$) are usually of poor quality and hardly spread uniformly in all directions of the state space, the encoder is not sufficiently trained in uncovered regions ($\mathcal{S}_{\text{late}}$), so the variational posterior q_ϕ cannot deviate from the prior $p(z)$. We validate this experimentally in Appendix B. This makes the KL divergence $\epsilon_{s,s'}$ extremely small in the uncovered state space, and thus the representation difference $\|\phi(s') - \phi(s)\|_2$ and the intrinsic reward approach zero. Therefore, we assume there exist constants $\epsilon_{\text{early}} \gg \epsilon_{\text{late}} > 0$ such that for any $s \in \mathcal{S}_{\text{late}}$, $D_{\mathrm{KL}}(q_\phi(z \mid s, s') \| p(z)) \le \epsilon_{\text{late}}$, and for any $s \in \mathcal{S}_{\text{early}}$, $D_{\mathrm{KL}}(q_\phi(z \mid s, s') \| p(z)) \le \epsilon_{\text{early}}$.

To further analyze the behavior of the optimal policy π^* under the derived intrinsic reward, we introduce the following definitions. Let $R_{\text{late}} = \sqrt{2d\epsilon_{\text{late}}} + C\epsilon_{\text{late}}$, $R_{\text{early}} = \sqrt{2d\epsilon_{\text{early}}} + C\epsilon_{\text{early}}$, where C is the constant in $O(\epsilon)$ in Prop. 2. For a policy π and skill z, we define the discounted transition occupancy measure as $d_z^\pi(s, s') = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s, s_{t+1} = s' \mid z)$, and define the corresponding state occupancy measure as $d_z^\pi(s) = \sum_{s'} d_z^\pi(s, s')$. Using this measure, we further define the state reward as $r(s, z) = \frac{1}{d_z^{\pi^*}(s)}[\sum_{s'} d_z^{\pi^*}(s, s') r(s, s', z)]\mathbb{I}[d_z^{\pi^*}(s) > 0]$, where π^* is the optimal policy under the reward induced by Prop. 2. Let π_b be the behavior policy used to collect data in $\mathcal{S}_{\text{early}}$, i.e., $\mathcal{S}_{\text{early}} = \{s : d^{\pi_b}(s) > 0\}$. Consider a state space partition: $\mathcal{S}_{\text{early},z} = \{s \in \mathcal{S}_{\text{early}} : r(s, z) \ge R_g\}$, $\mathcal{S}_{\text{early},-z} = \mathcal{S}_{\text{early}} \setminus \mathcal{S}_{\text{early},z}$, where R_g is a constant satisfying $R_{\text{late}} < R_g \le R_{\text{early}}$. Finally, we define the skill-conditioned state occupancy measure in the partitioned state subspaces: $q_{e,z} = d_z^{\pi_b}(\mathcal{S}_{\text{early},z})$, $p_l^* = d_z^{\pi^*}(\mathcal{S}_{\text{late}})$.

Proposition 3. *Under the reward induced by Prop. 2, the skill-conditioned state occupancy measure of the optimal policy in the uncovered state set is upper bounded by*

$$p_l^* \le \frac{2R_{\text{early}} - (R_g + R_{\text{early}})q_{e,z}}{R_{\text{early}} - R_{\text{late}}}. \tag{9}$$

Proposition 3 quantifies the optimal policy's state occupancy measures for S_{late}, denoted as p_l^*. According to Eq. 9, p_l^* is mainly influenced by the behavior policy for training data collection, as R_g and $q_{e,z}$ jointly reflect the alignment between the skill latent and the skill trajectory of the behavior policy. To improve the behavior policy, its state visitation should be concentrated on the region of the state space that is best aligned with the skill z (i.e., larger $q_{e,z}$) and where the intrinsic reward is larger (i.e., larger R_g). Therefore, as the behavior policy approaches the optimal policy, we have $q_{e,z}$ close to 1, and R_g close to R_{early}. We can assume $R_g = R_{\text{early}} - \delta$ for a small $\delta \ge 0$. Then

$$p_l^* \approx \frac{2R_{\text{early}} - (R_{\text{early}} - \delta + R_{\text{early}})q_{e,z}}{R_{\text{early}}} = 2 - (2 - \delta/R_{\text{early}})q_{e,z} = \delta/R_{\text{early}}, \tag{10}$$

which can be very small. This indicates that policy training and encoder training form a positive feedback process. As the behavior policy gradually approaches optimality on the collected data during training, the trajectories collected by the policy become increasingly aligned with the intrinsic reward conditioned on the same skill latent derived by the encoder. Consequently, the probability that the policy visits the $\mathcal{S}_{\text{late}}$ region and collects new data gradually decreases. When the behavior policy reaches the optimum under the current data S_{early}, being well-aligned with the intrinsic reward conditioned on the same skill latent, i.e., $R_g \to R_{\text{early}}$ and $q_{e,z} \to 1$, we have $p_l^* \to 0$, which means the local optimal policy will visit almost only the S_{early} region and avoid the S_{late} region, and

the algorithm completely loses its exploration ability. Since early data usually have poor coverage and quality, the policy can easily reach a local optimum early, which prevents further exploration in this direction and thus leads to the dead zone phenomenon.

Next, we show in Proposition 4 that once the skill encoder and the policy have converged on the early-epoch data $\mathcal{S}_{\text{early}}$, escaping from the local optimum is nearly impossible.

Proposition 4. *Consider that at time step t_0, the replay buffer $\mathcal{D}$ contains N transitions from $\mathcal{S}_{\text{early}}$, the encoder ϕ has converged to ϕ^*, and the policy π has converged to π^*. To ensure driving the encoder out of the ε-neighborhood of ϕ^*, i.e., $\|\phi - \phi^*\| > \varepsilon$, the required sample size m is lower bounded by*

$$m \geq \frac{1}{2}\left(\frac{Nc_e}{C_l} + \frac{\varepsilon}{\eta\beta C_l} + \sqrt{\left(\frac{Nc_e}{C_l} + \frac{\varepsilon}{\eta\beta C_l}\right)^2 + \frac{4\varepsilon N}{\eta\beta C_l}}\right), \tag{11}$$

where η is the learning rate, c_e is a lower bound on the magnitudes of the sample gradients for transitions from $\mathcal{S}_{\text{early}}$ against escape, and C_l is an upper bound on those for transitions from $\mathcal{S}_{\text{late}}$.

Proposition 4 further quantifies the minimum sample size required to escape this local optimum on limited data, i.e., to change the encoder parameters by ε, when both the policy and encoder have converged to the optimal behavior under $\mathcal{S}_{\text{early},z}$. If only a very small parameter change is needed, i.e., $\varepsilon \to 0$, the ε term in the lower bound of Proposition 4 can be ignored, yielding $m = \Omega\left(c_e/C_l \cdot N\right)$. This shows that even a very small encoder change requires new samples of the same order as the existing sample size. By Proposition 3, after the policy converges, the probability of exploring new regions approaches 0, so the time required to collect that many samples approaches infinity, and the policy cannot escape the converged suboptimal solution.

4 RASD: Reward-Aligned Skill Discovery

In this section, we propose a simple yet effective method to address the policy-reward mismatch problem by providing more on-policy data for policy and encoder updates. This prevents exploration suppression caused by delayed reward updates when the policy explores new regions, thereby enhancing early policy exploration.

4.1 Align intrinsic reward update with current policy

To address the policy-reward mismatch issue, we should ensure that the intrinsic reward encourages exploration. As analyzed in Section 3, the intrinsic reward depends strongly on the data coverage for skill encoder training (Prop. 2), and early data are implicitly assigned greater weight during training (Prop. 1). If the amount of early data far exceeds that of newly collected data (which may be very sparse), the discriminator can hardly update rewards for newly visited states. Instead, it almost only maintains rewards for those early states that have been sufficiently explored. Consequently, highly discriminative states newly explored do not receive the high rewards they deserve, thereby suppressing further exploration.

To ensure the intrinsic reward focuses on regions newly explored by the policy, the training data distribution of the encoder must significantly overlap with that of the current policy. A straightforward way to achieve this is to update the skill discriminator with on-policy transitions sampled by the current policy. However, this could lead to a severe issue with sample efficiency, as data cannot be reused after a policy update (Schulman et al., 2017).

Inspired by Zhang & Sutton (2018), we therefore consider a trade-off between the off-policy update method in existing SD methods and the on-policy update method. In addition to the existing off-policy replay buffer for discriminator update, we introduce an additional recent-policy replay buffer that stores only more up-to-date data compared with the off-policy buffer. When sampling data for gradient calculation, we draw from the recent-policy buffer with probability $p \in (0, 1)$ and from the off-policy buffer with probability $1 - p$. The derived transitions here are a mixture of recent-policy and off-policy transitions, which emphasize the recent-policy transitions for in-time reward updates while preserving off-policy transitions to ensure high sample efficiency.

The recent-policy buffer is achieved using a first-in-first-out (FIFO) buffer with a much smaller capacity than the off-policy buffer. Compared with pure on-policy samples, this recent-policy buffer

Algorithm 1 RASD

Require: Recent-policy buffer sample rate p, recent-policy buffer size N_{on}, off-policy buffer size N_{off}, mini-batch size b, total epoch number E

1: Initialize recent-policy replay buffer $\mathcal{D}_{\text{on}}$ of size N_{on}
2: Initialize off-policy replay buffer $\mathcal{D}_{\text{off}}$ of size N_{off}
3: **for** each epoch $e = 1, 2 \ldots, E$ **do**
4: Sample skill $z \sim p(z)$, rollout with policy $\pi(a|s, z)$ and store (s, a, s', z) into $\mathcal{D}_{\text{off}}$ and $\mathcal{D}_{\text{on}}$
5: Remove old transitions from $\mathcal{D}_{\text{off}}$ and $\mathcal{D}_{\text{on}}$ if the number of transitions in the buffer exceeds the corresponding capacity N_{off} and N_{on}
6: Sample bp transitions from $\mathcal{D}_{\text{on}}$ and $b(1-p)$ transitions from $\mathcal{D}_{\text{off}}$
7: Update the skill encoder $\phi(s)$ to maximize Eq. 3 using all sampled b transitions
8: Update the Lagrange multiplier λ to minimize Eq. 4 using all sampled b transitions
9: Update the Q-function and the policy $\pi(a|s, z)$ with SAC (Haarnoja et al., 2018) using all sampled b transitions and the intrinsic reward $r(s, s', z) = (\phi(s') - \phi(s))^\top z$
10: **end for**

is less aligned with the current policy, as transitions are not immediately discarded after policy update. However, as the target is to let the discriminator focus on the newly explored area by the policy, such recent-policy samples are still useful because they are also near the exploration frontier of the policy. In addition, using more recent-policy samples helps stabilize the gradient updates of the discriminator, as pure on-policy samples may yield noisy and high-variance gradients.

4.2 Align skill update with current intrinsic reward

As the discriminator and the policy update alternately in skill discovery methods, the policy faces a similar problem as the discriminator. Specifically, most skill discovery methods use an off-policy RL algorithm for policy updates, which involves a Q-function that accumulates the impacts of intrinsic rewards along skill trajectories:

$$Q^\pi(s, a, z) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \left[\gamma^t r(s_t, s_{t+1}, z) \right] \middle| s_0 = s, a_0 = a \right], \tag{12}$$

where γ is the discount factor. If we only augment the discriminator update with an additional recent-policy buffer and leave the policy update unchanged, though the intrinsic reward could be accurate on the exploration frontier, the Q-function is still trapped in a large amount of old data that the policy has already mastered.

Therefore, the policy update should also be augmented with a recent-policy buffer, same as the discriminator, so that the Q-function can focus more on the exploration frontier. Note that, different from the discriminator, the Q-function update requires the temporal difference (TD) error between subsequent states (s, s'). This means that the Q-functions of all states leading to the exploration frontier must be estimated accurately before that of states on the exploration frontier can be accurately estimated. Therefore, pure on-policy updates for the policy are almost infeasible for skill discovery, which justifies the preservation of the off-policy buffer in RASD.

4.3 Implementation Details

The overall framework of RASD is provided in Algorithm 1. Our method only introduces two new hyperparameters, namely the recent-policy buffer sample rate p and the recent-policy buffer size N_{on}. Increasing p and decreasing N_{on} can help ensure a high sampling probability for data newly explored by the policy, thereby alleviating the policy-reward mismatch issue. However, if p is too large, the sampled transitions will consist mostly of on-policy data, which will degrade performance as the policy may forget previous behaviors.

In addition, our method is easy to implement, incurs negligible additional computational cost, and introduces only a small additional memory overhead. Specifically, RASD requires only an additional recent-policy buffer. As this buffer stores only the most recent data, the additional memory overhead is low, i.e., $1/10$ of the off-policy buffer in our experiments.

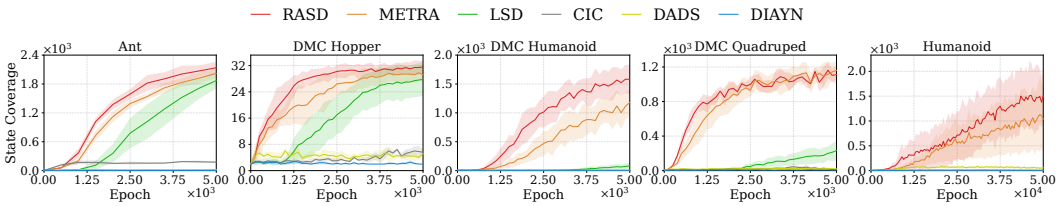


Figure 2: State coverage of all algorithms across all training epochs. We report the mean coverage (solid line) and its standard deviation (dashed area) across 5 seeds.

Table 1: Zero-shot downstream task performance in the goal-reaching task.

Method	DMC Humanoid	Hopper
RASD	0.888 $\pm$ 0.090	0.988 $\pm$ 0.011
METRA	0.700 $\pm$ 0.139	0.904 $\pm$ 0.040
LSD	0.236 $\pm$ 0.163	0.960 $\pm$ 0.089
DIAYN	0.088 $\pm$ 0.033	0.348 $\pm$ 0.122

Table 2: Hierarchical control downstream task performance in the goal-reaching task.

Method	DMC Humanoid	Hopper
RASD	61.500 $\pm$ 17.742	125.050 $\pm$ 10.508
METRA	53.338 $\pm$ 29.156	123.329 $\pm$ 11.297
LSD	40.354 $\pm$ 17.355	123.592 $\pm$ 16.852
DIAYN	16.667 $\pm$ 9.462	29.388 $\pm$ 12.550

5 Experiment

In this section, we conduct extensive experiments to answer the following questions: *Q1*: Does RASD learn useful skills that achieve broad state-space coverage and benefit downstream tasks? *Q2*: Does RASD effectively mitigate the policy-reward mismatch issue and the dead-zone phenomenon? *Q3*: What is the individual contribution of each proposed technique in RASD?

5.1 Setup

Environments. We evaluate RASD on five complex robotic locomotion environments, including Ant and Humanoid from OpenAI Gym (Todorov et al., 2012; Brockman et al., 2016), and Hopper, Quadruped, and DMC Humanoid from DMControl (Tassa et al., 2018). Further details on these environments are provided in Appendix D.1.

Baselines and implementation. To evaluate RASD, we compare RASD against five representative skill discovery algorithms, including DIAYN (Eysenbach et al., 2019), CIC (Laskin et al., 2022), DADS (Sharma et al., 2020), three MI-based skill discovery methods; and LSD (Park et al., 2022) and METRA (Park et al., 2024), two DSD methods using Euclidean distance and Temporal distance as the distance measure, respectively. Further details are provided in Appendix D.3.

Metrics. Following Park et al. (2024), we measure the performance of RASD using two metrics. (1) **State coverage** measures the agent's ability to explore the state space. This metric computes state coverage by counting the unique 1×1 x-y bins visited by the agent (or 1-unit x-axis bins for Hopper). (2) **Downstream task performance** evaluates the utility of learned skills in goal-reaching downstream tasks. We consider both a zero-shot setting where the learned skills are directly reused and a task-specific hierarchical control setting where the learned skills are frozen and an additional high-level controller is trained to optimize performance on the downstream task. Further details on the downstream tasks are provided in Appendix D.2.

5.2 Main Results

State coverage. To evaluate the quality of the learned skills, we first assess their state space coverage. Fig. 2 reports the state coverage of RASD and all baselines throughout the skill discovery process. As shown in the figure, RASD improves skill coverage more rapidly than the baselines and achieves better final performance in most environments. Notably, the performance advantage of RASD is much higher in the more complex Humanoid environments, demonstrating the superior exploration ability of the proposed method in challenging tasks.

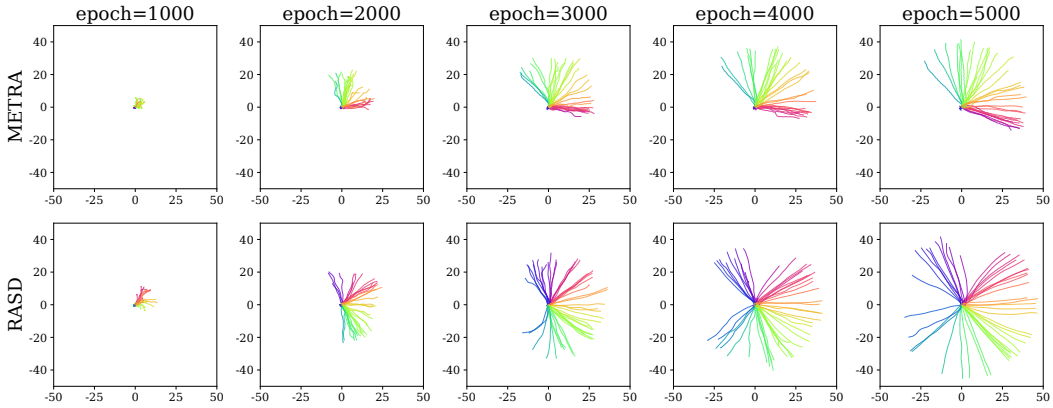


Figure 3: Visualizations of skills learned by RASD and METRA across different training epochs on the DMC Humanoid task. We plot x-y trajectories sampled from the learned policies. Different colors represent distinct skills z. More illustrations are provided in Fig. 7.

Downstream Task Performance. To evaluate the utility of the skills learned by RASD for downstream tasks, we conducted experiments on goal-reaching downstream tasks, assessing both the zero-shot performance of the algorithm and its performance as a fixed low-level policy in a hierarchical controller. Tables 1 and 2 show the downstream performance of RASD and baselines. As shown in these tables, RASD outperforms all baselines in most downstream tasks, indicating that its learned skills perform well when directly applied to downstream tasks and can serve as a high-quality low-level policy for a hierarchical controller.

Visualization. To illustrate the dead-zone phenomenon, we visualize the trajectories of the skills learned by RASD and METRA on the DMC Humanoid task in Fig. 3. The trajectories for both methods are generated using the same random seed, ensuring identical initializations of the encoders and policies. As shown in Fig. 3, in the early epochs of skill discovery, the skills learned by both RASD and METRA are trapped in a narrow range of angles. However, as training progresses, RASD gradually expands the angular coverage of its learned skills and eventually nearly covers the entire xy-plane, whereas METRA covers only half of the plane, leaving the other half of the plane as the dead zone. This shows that RASD can effectively mitigate the dead zone phenomenon.

5.3 Ablation Study

Smaller buffer vs. dual buffer. Compared with existing SD methods, RASD uses an additional recent-policy buffer to address the policy-reward mismatch issue. We therefore examine whether the performance gain comes from the smaller recent-policy buffer or from our dual buffer mechanism that combines both buffers. To this end, we remove the off-policy buffer from RASD and use only the recent-policy buffer. Fig. 4 (left) shows that the variant using only the recent-policy buffer underperforms RASD. Nevertheless, this variant still yields a small improvement over METRA, possibly due to more timely updates of the intrinsic reward. This shows that early data are not entirely useless. Restricting the buffer to recent-policy data limits the data available for policy and encoder learning, causing catastrophic forgetting and thus degrading performance.

Ablation of recent-policy buffer sample rate. The sample rate determines the fraction of data sampled from the recent-policy buffer. Note that a ratio of 1.0 means training with only the recent-policy buffer. Fig. 4 (middle) shows that increasing the rate improves performance, but an excessively large rate degrades performance due to catastrophic forgetting in the policy.

Ablation of recent-policy buffer size. The recent-policy buffer size affects the distribution difference between the data in the recent-policy buffer and the current policy. The smaller the capacity, the better the sampled data aligns with the current policy. Fig. 4 (right) shows that reducing the buffer size improves performance, possibly due to more timely updates of the intrinsic reward, but an excessively small buffer size provides limited improvement, as 100k and 32k perform similarly.

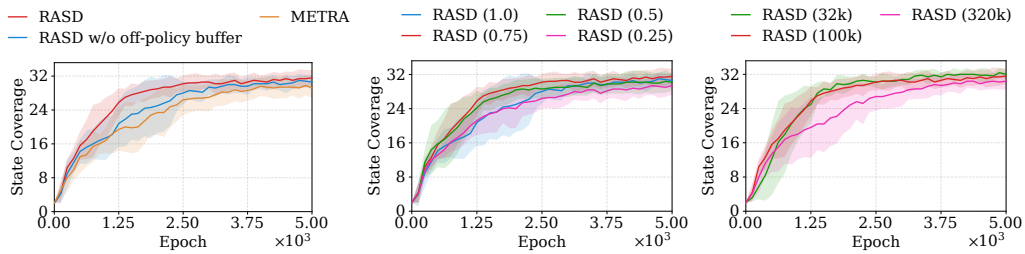


Figure 4: State coverage of RASD across all training epochs on DMC Hopper for ablation. (1) Left figure: we evaluate RASD when removing the off-policy buffer. (2) Middle figure: we evaluate RASD with different recent-policy buffer sample rates (values in brackets). (3) Right figure: we evaluate RASD with different recent-policy buffer sizes (values in brackets).

Recent-Policy Buffer in Encoder vs. Policy. We compare RASD with two variants: RASD (reward), which applies the dual buffer only to the skill encoder, and RASD (policy), which applies it only to the policy and the Q-function. As shown in Fig. 8, both variants reduce state coverage compared with full RASD, justifying the necessity of using the recent-policy buffer in both modules.

6 Related Work

Unsupervised skill discovery. Skill discovery (SD) is a URL method that constructs intrinsic rewards with empowerment measures (Xie et al., 2022). The target is to acquire task-agnostic environment knowledge as reusable skills to facilitate downstream tasks. A typical empowerment measure is the mutual information (MI) between state s and skill latent z (Eysenbach et al., 2019; Gregor et al., 2016; Sharma et al., 2020; Liu & Abbeel, 2021; Laskin et al., 2022; Zheng et al., 2025). However, MI can lead to static behaviors in complex tasks (Park et al., 2022; 2024), which limits the utility of the learned skills. Distance-maximizing skill discovery (DSD) methods are representative approaches for addressing this static-behavior issue (Park et al., 2023; 2024), as they encourage state-space coverage by aligning latent-space distances with state-space distances. Typical distance metrics include Euclidean distance (Park et al., 2022), the negative log-likelihood of an estimated transition probability (Park et al., 2023), and the temporal distance (Park et al., 2024). However, even DSD methods cannot fully cover the skill space, leaving dead zones in the state space uncovered.

Experience replay. The replay buffer enables off-policy algorithms to reuse data generated by past policies, thereby improving sample efficiency (Lin, 1992). However, under uniform sampling, abundant old data can hinder the sampling of useful data (Zhang & Sutton, 2018). To address this, existing methods prioritize samples in the replay buffer, making useful samples more likely to be used for policy training (Korniak et al., 2026). Typical priority criteria include temporal difference (Schaul et al., 2016), regret (Liu et al., 2021), and Q-function reliability (Pleiss et al., 2025). Beyond assigning priorities to data, some works collect or generate useful data to augment the replay buffer (Andrychowicz et al., 2018; Sinha et al., 2022). Unlike these methods, which focus on policy training and improve off-policy sampling by emphasizing useful data, RASD targets the policy-reward mismatch problem in intrinsic reward learning and leverages the off-policy buffer as a supplementary mechanism to mitigate catastrophic forgetting.

7 Conclusion

This paper focuses on the dead zone phenomenon in skill discovery. Through theoretical analysis, we characterize the evolving behavior of the intrinsic reward and the policy during skill discovery, and identify that a core issue is policy-reward mismatch, i.e., the intrinsic reward fails to encourage exploration, instead trapping the policy in a local optimum and suppressing further exploration. To address this issue, we propose RASD, a simple yet efficient method that provides more on-policy data for the policy and encoder to ensure quick reward adaptation. Extensive experiments show that our method effectively improves the state-space coverage of skills learned by existing skill discovery methods and achieves better performance on downstream tasks.

AI USE STATEMENT

In this work, we used generative AI tools to edit a research paper to improve readability. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets, assist with translation, clean and reformat datasets are not applicable to this work. We have reviewed all AI-assisted work. The authors have proofread the final manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Detailed proofs for the propositions are provided in Appendix A. The code implementing of RASD is provided in the supplementary material. The pseudocode for RASD is presented in Algorithm 1. Implementation details, including environment setup, evaluation metrics, and hyperparameters, are provided in Appendix D.3.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay, 2018. URL https://arxiv.org/abs/1707.01495.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Benjamin Eysenbach, Julian Ibarz, Abhishek Gupta, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control, 2016. URL https://arxiv.org/abs/1611.07507.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018.

Zilin Kang, Chenyuan Hu, Yu Luo, Zhecheng Yuan, Ruijie Zheng, and Huazhe Xu. A forget-and-grow strategy for deep reinforcement learning scaling in continuous control. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=VhmTXbsdtx.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Michal Korniak, Mikołaj Czarnecki, Yarden As, Piotr Miłoś, Pieter Abbeel, and Michal Nauman. When does non-uniform replay matter in reinforcement learning?, 2026. URL https://arxiv.org/abs/2605.10236.

Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. *Advances in Neural Information Processing Systems*, 35:34478–34491, 2022.

Long Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.*, 8:293–321, 1992. doi: 10.1007/BF00992699. URL https://doi.org/10.1007/BF00992699.

Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pp. 6736–6747. PMLR, 2021.

Xu-Hui Liu, Zhenghai Xue, Jingcheng Pang, Shengyi Jiang, Feng Xu, and Yang Yu. Regret minimization experience replay in off-policy reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17604–17615. Curran Associates, Inc., 2021.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations*, 2022.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 27225–27245, 2023.

Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. In *International Conference on Learning Representations*, 2024.

Leonard S. Pleiss, Tobias Sutter, and Maximilian Schiffer. Reliability-adjusted prioritized experience replay, 2025. URL https://arxiv.org/abs/2506.18482.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016. URL https://arxiv.org/abs/1511.05952.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. In Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer (eds.), *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pp. 110–123. PMLR, 23–24 Jun 2022.

DJ Strouse, Kate Baumli, David Warde-Farley, Volodymyr Mnih, and Steven Stenberg Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cU8rknuhxc.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Zhihui Xie, Zichuan Lin, Junyou Li, Shuai Li, and Deheng Ye. Pretraining in deep reinforcement learning: A survey. *arXiv preprint arXiv:2211.03959*, 2022.

Shangtong Zhang and Richard S. Sutton. A deeper look at experience replay, 2018. URL https://arxiv.org/abs/1712.01275.

Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a MISL fly? analysis and ingredients for mutual information skill learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

A PROOF

A.1 PROPOSITION 1

Proposition 1. *Consider a replay buffer $\mathcal{D}$ storing N transitions. At each environment step, k gradient updates are performed, and each gradient update uses a mini-batch of size b sampled uniformly from $\mathcal{D}$. Let $\beta = bk$. For the t-th transition added to the buffer $(1 < t \leq N)$, the expected number of times it is sampled during training, $\mathbb{E}[n_t]$, satisfies*

$$\mathbb{E}[n_t] = \Theta\left(\log \frac{N}{t}\right). \tag{6}$$

Proof. In a replay buffer with uniform sampling, when the buffer contains i transitions, the probability that any particular transition is sampled is $1/i$.

Therefore, from the time the t-th transition is added to the buffer, i.e., $i = t$, until the end of training, i.e., $i = N$, the expected number of times it is sampled is

$$\mathbb{E}[n_t] = \beta \sum_{i=t}^{N} \frac{1}{i} = \beta(H_N - H_{t-1}), \tag{13}$$

where $H_n = \sum_{j=1}^{n} \frac{1}{j}$.

Using the inequalities for harmonic numbers $\log(n+1) < H_n < \log(n) + 1$, we have

$$\mathbb{E}[n_t] > \beta\left(\log(N+1) - \log(t-1) - 1\right) = \beta\left(\log \frac{N+1}{t-1} - 1\right), \tag{14}$$

$$\mathbb{E}[n_t] < \beta\left(\log(N) + 1 - \log(t)\right) = \beta\left(\log \frac{N}{t} + 1\right). \tag{15}$$

Therefore, when $N \gg t$, $\mathbb{E}[n_t] = \Theta\left(\log \frac{N}{t}\right)$. □

The proof of Proposition 1 is inspired by Kang et al. (2025).

A.2 PROPOSITION 2

Proposition 2. *For any transition pair (s, s'), if the variational posterior satisfies $D_{\mathrm{KL}}(q_\phi(z \mid s, s') \| p(z)) \leq \epsilon_{s,s'}$, where $\epsilon_{s,s'} \geq 0$, $p(z) = \mathrm{UNIF}(\mathbb{S}^{d-1})$ (d is the dimension of the skill latent z), then the norm of the representation difference satisfies*

$$\|\phi(s') - \phi(s)\|_2 \leq \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}), \tag{7}$$

and the intrinsic reward satisfies

$$\sup_{z \in \mathbb{S}^{d-1}} |r(s, s', z)| \leq \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}). \tag{8}$$

In particular, if $\epsilon_{s,s'} = 0$, then the posterior cannot deviate from the prior, so $\phi(s') = \phi(s)$, and for all $z \in \mathbb{S}^{d-1}$ we have $r(s, s', z) = 0$.

Proof. By the definition of $q_\phi(z \mid s, s')$ and $p(z) = \mathrm{UNIF}(\mathbb{S}^{d-1})$, for $\delta_{s,s'} = \phi(s') - \phi(s)$, the variational distribution can be written as

$$q_\phi(z \mid s, s') = \frac{p(z)\exp(\delta_{s,s'}^\top z)}{Z(\delta_{s,s'})}, \quad Z(\delta_{s,s'}) = \mathbb{E}_{p(z')}[\exp(\delta_{s,s'}^\top z')]. \tag{16}$$

This is the standard von Mises-Fisher distribution $\mathrm{vMF}(\mu, \kappa_{s,s'})$, with mean direction $\mu = \delta_{s,s'}/\|\delta_{s,s'}\|_2$ and concentration parameter $\kappa_{s,s'} = \|\delta_{s,s'}\|_2$.

By Lemma 1, when $\kappa_{s,s'} \to 0$, we have

$$D_{KL}(q_\phi(z \mid s, s') \| p(z)) = \frac{\kappa_{s,s'}^2}{2d} + O(\kappa_{s,s'}^4). \tag{17}$$

Therefore, when $D_{KL} \leq \epsilon_{s,s'}$ and $\epsilon_{s,s'}$ is sufficiently small, solving the inequality yields

$$\kappa_{s,s'} = \|\delta_{s,s'}\|_2 \leq \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}). \tag{18}$$

Using the definition of the reward function,

$$|r(s, s', z)| = |\delta_{s,s'}^\top z| \leq \|\delta_{s,s'}\|_2 \|z\|_2 = \|\delta_{s,s'}\|_2, \tag{19}$$

we have

$$\sup_{z \in \mathbb{S}^{d-1}} |r(s, s', z)| \leq \sqrt{2d \cdot \epsilon_{s,s'}} + O(\epsilon_{s,s'}). \tag{20}$$

If $\epsilon_{s,s'} = 0$, then $D_{KL}(q_\phi(z \mid s, s') \| p(z)) = 0$. For the standard vMF distribution, $D_{KL}(\text{vMF}(\mu, \kappa_{s,s'}) \| \text{UNIF}(\mathbb{S}^{d-1})) = 0$ is equivalent to $\kappa_{s,s'} = 0$, which means $\delta_{s,s'} = \mathbf{0}$, $\phi(s') = \phi(s)$. Therefore $r(s, s', z) = \mathbf{0}^\top z = 0$ for all z, and the skill discrimination is identically zero. This completes the proof. □

A.3 Proposition 3

For a policy π and skill z, define the discounted transition occupancy measure and state occupancy measure as

$$d_z^\pi(s, s') = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s, s_{t+1} = s' \mid z), \tag{21}$$

$$d_z^\pi(s) = \sum_{s'} d_z^\pi(s, s') = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s \mid z). \tag{22}$$

Define the state reward as

$$r(s, z) = \frac{\sum_{s'} d_z^{\pi^*}(s, s') r(s, s', z)}{d_z^{\pi^*}(s)} \mathbb{I}[d_z^{\pi^*}(s) > 0]. \tag{23}$$

Let $\mathcal{S}_{\text{early}}$ and $\mathcal{S}_{\text{late}}$ be a partition of the state space $\mathcal{S}$ with $\mathcal{S}_{\text{late}} = \mathcal{S} \setminus \mathcal{S}_{\text{early}}$. Let π_b be the behavior policy used to collect data in $\mathcal{S}_{\text{early}}$, and let π^* be the optimal policy under the reward induced by Prop. 2. Let $\gamma \in [0, 1)$ be the discount factor.

Assume there exist constants $\epsilon_{\text{early}} \gg \epsilon_{\text{late}} > 0$ such that for any $s \in \mathcal{S}_{\text{late}}$,

$$D_{\text{KL}}(q_\phi(z \mid s, s') \| p(z)) \leq \epsilon_{\text{late}}, \tag{24}$$

and for any $s \in \mathcal{S}_{\text{early}}$,

$$D_{\text{KL}}(q_\phi(z \mid s, s') \| p(z)) \leq \epsilon_{\text{early}}. \tag{25}$$

By Prop. 2, define

$$R_{\text{late}} = \sqrt{2d\epsilon_{\text{late}}} + C\epsilon_{\text{late}}, \quad R_{\text{early}} = \sqrt{2d\epsilon_{\text{early}}} + C\epsilon_{\text{early}}, \tag{26}$$

where C is the constant in $O(\epsilon)$ in Prop. 2. Then $R_{\text{early}} \gg R_{\text{late}}$. The relation $\epsilon_{\text{early}} \gg \epsilon_{\text{late}} > 0$ holds because the encoder has been sufficiently trained on $\mathcal{S}_{\text{early}}$ but not on $\mathcal{S}_{\text{late}}$, so the prior and posterior differ significantly on $\mathcal{S}_{\text{early}}$ and only slightly on $\mathcal{S}_{\text{late}}$.

Consider a constant R_g satisfying $R_{\text{late}} < R_g \leq R_{\text{early}}$. We further partition $\mathcal{S}_{\text{early}}$ as follows:

$$\mathcal{S}_{\text{early},z} = \{s \in \mathcal{S}_{\text{early}} : r(s, z) \geq R_g\}, \quad \mathcal{S}_{\text{early},-z} = \mathcal{S}_{\text{early}} \setminus \mathcal{S}_{\text{early},z}. \tag{27}$$

We then define the following state occupancy measures for the above partition:

$$\begin{aligned} q_{e,z} &= d_z^{\pi_b}(\mathcal{S}_{\text{early},z}), & q_{e,-z} &= d_z^{\pi_b}(\mathcal{S}_{\text{early},-z}), \\ p_l^* &= d_z^{\pi^*}(\mathcal{S}_{\text{late}}), & p_e^* &= d_z^{\pi^*}(\mathcal{S}_{\text{early}}) = 1 - p_l^*, \\ p_{e,z}^* &= d_z^{\pi^*}(\mathcal{S}_{\text{early},z}), & p_{e,-z}^* &= d_z^{\pi^*}(\mathcal{S}_{\text{early},-z}). \end{aligned} \tag{28}$$

We assume that there exists z such that $\mathcal{S}_{\text{early},z}$ is nonempty. This requires that on some state regions the encoder prior and posterior differ significantly, which should hold when the encoder is sufficiently trained. The following discussion is restricted to z with $\mathcal{S}_{\text{early},z}$ nonempty.

Proposition 3. *Under the reward induced by Prop. 2, the skill-conditioned state occupancy measure of the optimal policy in the uncovered state set is upper bounded by*

$$p_l^* \leq \frac{2R_{\text{early}} - (R_g + R_{\text{early}})q_{e,z}}{R_{\text{early}} - R_{\text{late}}}. \tag{9}$$

Proof. By Prop. 2, the state reward $r(s,z)$ satisfies the following bounds. For $s \in \mathcal{S}_{\text{late}}$, $-R_{\text{late}} \leq r(s,z) \leq R_{\text{late}}$. For $s \in \mathcal{S}_{\text{early}}$, $-R_{\text{early}} \leq r(s,z) \leq R_{\text{early}}$. For $s \in \mathcal{S}_{\text{early},z}$, $R_g \leq r(s,z) \leq R_{\text{early}}$. For $s \in \mathcal{S}_{\text{early},-z}$, $-R_{\text{early}} \leq r(s,z) \leq R_g$.

Since π_b is the behavior policy used to collect data in $\mathcal{S}_{\text{early}}$, it satisfies $\mathcal{S}_{\text{early}} = \{s : d_z^{\pi_b}(s) > 0\}$, $d_z^{\pi_b}(\mathcal{S}_{\text{late}}) = 0$, and $q_{e,-z} = 1 - q_{e,z}$. The behavior policy return measured by $r(s,z)$ is

$$J(\pi_b; z) = \sum_{s \in \mathcal{S}_{\text{early},z}} d_z^{\pi_b}(s) r(s,z) + \sum_{s \in \mathcal{S}_{\text{early},-z}} d_z^{\pi_b}(s) r(s,z). \tag{29}$$

Using the lower bounds for the rewards in the two regions,

$$\begin{aligned} J(\pi_b; z) &\geq R_g q_{e,z} - R_{\text{early}} q_{e,-z} \\ &= R_g q_{e,z} - R_{\text{early}}(1 - q_{e,z}) \\ &= (R_g + R_{\text{early}}) q_{e,z} - R_{\text{early}}. \end{aligned} \tag{30}$$

The optimal policy return is

$$J(\pi^*; z) = \sum_s d_z^{\pi^*}(s) r(s,z). \tag{31}$$

Using the upper bounds for the rewards in the regions,

$$\begin{aligned} J(\pi^*; z) &\leq R_{\text{early}} p_{e,z}^* + R_g p_{e,-z}^* + R_{\text{late}} p_l^* \\ &\leq R_{\text{early}}(p_{e,z}^* + p_{e,-z}^*) + R_{\text{late}} p_l^* \\ &= R_{\text{early}} p_e^* + R_{\text{late}} p_l^*. \end{aligned} \tag{32}$$

Since $p_e^* = 1 - p_l^*$, we have

$$J(\pi^*; z) \leq R_{\text{early}} - (R_{\text{early}} - R_{\text{late}}) p_l^*. \tag{33}$$

By the optimality of π^*, $J(\pi^*; z) \geq J(\pi_b; z)$. Combining Eq. 32 and Eq. 30, we have

$$R_{\text{early}} - (R_{\text{early}} - R_{\text{late}}) p_l^* \geq (R_g + R_{\text{early}}) q_{e,z} - R_{\text{early}}. \tag{34}$$

Therefore,

$$p_l^* \leq \frac{2R_{\text{early}} - (R_g + R_{\text{early}}) q_{e,z}}{R_{\text{early}} - R_{\text{late}}}, \tag{35}$$

which concludes the proof. □

Corollary 1. *Under the same condition of Proposition 3, we have*

$$\frac{p_e^*}{p_l^*} \geq \frac{(R_g + R_{\text{early}}) q_{e,z} - R_{\text{early}}}{2R_{\text{early}} - (R_g + R_{\text{early}}) q_{e,z}}. \tag{36}$$

Proof. Let the right side of Eq. 35 be B. Since $p_e^* = 1 - p_l^* \geq 1 - B$, we have

$$\frac{p_e^*}{p_l^*} \geq \frac{1 - B}{B}. \tag{37}$$

Substituting the expression for B, we have

$$\begin{aligned} \frac{1 - B}{B} &= \frac{R_{\text{early}} - R_{\text{late}} - [2R_{\text{early}} - (R_g + R_{\text{early}}) q_{e,z}]}{2R_{\text{early}} - (R_g + R_{\text{early}}) q_{e,z}} \\ &= \frac{(R_g + R_{\text{early}}) q_{e,z} - R_{\text{early}} - R_{\text{late}}}{2R_{\text{early}} - (R_g + R_{\text{early}}) q_{e,z}}. \end{aligned} \tag{38}$$

Therefore,

$$\frac{d_z^{\pi^*}(\mathcal{S}_{\text{early}})}{d_z^{\pi^*}(\mathcal{S}_{\text{late}})} \geq \frac{(R_g + R_{\text{early}})q_{e,z} - R_{\text{early}} - R_{\text{late}}}{2R_{\text{early}} - (R_g + R_{\text{early}})q_{e,z}}. \tag{39}$$

Using $R_{\text{early}} \gg R_{\text{late}}$, this simplifies to

$$\frac{d_z^{\pi^*}(\mathcal{S}_{\text{early}})}{d_z^{\pi^*}(\mathcal{S}_{\text{late}})} \geq \frac{(R_g + R_{\text{early}})q_{e,z} - R_{\text{early}}}{2R_{\text{early}} - (R_g + R_{\text{early}})q_{e,z}}, \tag{40}$$

which concludes the proof. □

A.4 Proposition 4

Proposition 4. *Consider that at time step t_0, the replay buffer $\mathcal{D}$ contains N transitions from $\mathcal{S}_{\text{early}}$, the encoder ϕ has converged to ϕ^*, and the policy π has converged to π^*. To ensure driving the encoder out of the ε-neighborhood of ϕ^*, i.e., $\|\phi - \phi^*\| > \varepsilon$, the required sample size m is lower bounded by*

$$m \geq \frac{1}{2}\left(\frac{Nc_e}{C_l} + \frac{\varepsilon}{\eta\beta C_l} + \sqrt{\left(\frac{Nc_e}{C_l} + \frac{\varepsilon}{\eta\beta C_l}\right)^2 + \frac{4\varepsilon N}{\eta\beta C_l}}\right), \tag{11}$$

where η is the learning rate, c_e is a lower bound on the magnitudes of the sample gradients for transitions from $\mathcal{S}_{\text{early}}$ against escape, and C_l is an upper bound on those for transitions from $\mathcal{S}_{\text{late}}$.

Proof. Consider the best case, where all newly collected samples are far from $\mathcal{S}_{\text{early}}$, and the resulting gradients are aligned along a certain direction v and have the largest possible magnitude.

Then v is the final escape direction, and we have

$$v = \frac{\theta_{t_0+m} - \theta^*}{\|\theta_{t_0+m} - \theta^*\|}, \tag{41}$$

where θ is the parameter of ϕ. Consider the projection of the parameters onto direction v:

$$p_t = \langle \theta_t - \theta^*, v \rangle. \tag{42}$$

If finally $\|\theta_{t_0+m} - \theta^*\| > \varepsilon$, then necessarily $p_{t_0+m} > \varepsilon$.

Let $g(\theta)$ be the sample gradient. The single step update of projection p is

$$p' = \langle \theta - \eta g(\theta) - \theta^*, v \rangle = p - \eta \langle g(\theta), v \rangle. \tag{43}$$

For existing samples, by the assumption that transitions from $\mathcal{S}_{\text{early}}$ have sample gradient magnitude lower bounded by c_e against the escape direction, we have

$$p' \leq p - \eta c_e. \tag{44}$$

For newly collected samples, by the assumption that the sample gradient magnitude during training is upper bounded by C_l, in the best case its gradient projection is $-C_l$, and the update increases p the most:

$$p' \leq p + \eta C_l. \tag{45}$$

Therefore, during training from t_0 to $t_0 + m$, if T_e updates use existing samples and T_l updates use newly collected samples, then

$$p_{t_0+m} \leq \eta C_l T_l - \eta c_e T_e. \tag{46}$$

Considering the escape condition $p_{t_0+m} > \varepsilon$, we have

$$C_l T_l - c_e T_e > \frac{\varepsilon}{\eta}. \tag{47}$$

This inequality holds for the actual update counts T_l, T_e in the random sampling process. Taking expectations, a necessary condition is

$$C_l \mathbb{E}[T_l] - c_e \mathbb{E}[T_e] > \frac{\varepsilon}{\eta}. \tag{48}$$

Next, we compute the expected sampling counts $\mathbb{E}[T_l]$ and $\mathbb{E}[T_e]$.

After t_0, the buffer size gradually increases from N to $N+m$. For any early sample, during the increase of the buffer size from $N+1$ to $N+m$, its expected number of times being sampled is

$$\beta \sum_{i=N+1}^{N+m} \frac{1}{i} = \beta \left(H_{N+m} - H_N \right). \tag{49}$$

There are N early samples, so

$$\mathbb{E}[T_e] = \beta N \left(H_{N+m} - H_N \right). \tag{50}$$

When the j-th late sample is added, the buffer size is $N+j$. Its expected number of times being sampled from insertion to t_0+m is

$$\beta \sum_{i=N+j}^{N+m} \frac{1}{i} = \beta \left(H_{N+m} - H_{N+j-1} \right). \tag{51}$$

Therefore, the total expected sampling count of all m late samples is

$$\begin{aligned} \mathbb{E}[T_l] &= \beta \sum_{j=1}^{m} \left(H_{N+m} - H_{N+j-1} \right) = \beta \sum_{i=N+1}^{N+m} \frac{i-N}{i} \\ &= \beta \left[m - N \left(H_{N+m} - H_N \right) \right]. \end{aligned} \tag{52}$$

Finally, we substitute the above sampling counts into $C_l \mathbb{E}[T_l] - c_e \mathbb{E}[T_e] > \varepsilon/\eta$, deriving

$$C_l \cdot \beta \left[m - N \left(H_{N+m} - H_N \right) \right] - c_e \cdot \beta N \left(H_{N+m} - H_N \right) > \frac{\varepsilon}{\eta}. \tag{53}$$

Rearranging and taking the closure because escape requires a strict inequality gives

$$m \geq \frac{\varepsilon}{\eta \beta C_l} + N \left(1 + \frac{c_e}{C_l} \right) \left(H_{N+m} - H_N \right). \tag{54}$$

Using the lower bound for the difference of harmonic numbers,

$$H_{N+m} - H_N = \sum_{i=N+1}^{N+m} \frac{1}{i} \geq m \cdot \frac{1}{N+m} = \frac{m}{N+m}. \tag{55}$$

Substituting this into Eq. 54 yields

$$m \geq \frac{\varepsilon}{\eta \beta C_l} + N \left(1 + \frac{c_e}{C_l} \right) \frac{m}{N+m}. \tag{56}$$

To make the above inequality an explicit bound, let

$$A = \frac{\varepsilon}{\eta \beta C_l}, \qquad B = 1 + \frac{c_e}{C_l}. \tag{57}$$

The above inequality simplifies to

$$m^2 + m \left(N - BN - A \right) - AN \geq 0. \tag{58}$$

Since $N - BN = N - N \left(1 + \frac{c_e}{C_l} \right) = -\frac{N c_e}{C_l}$, we can eliminate B, and derive

$$m^2 - m \left(\frac{N c_e}{C_l} + A \right) - AN \geq 0. \tag{59}$$

The positive root of this quadratic inequality is

$$m \geq \frac{\left(\frac{N c_e}{C_l} + A \right) + \sqrt{\left(\frac{N c_e}{C_l} + A \right)^2 + 4AN}}{2}. \tag{60}$$

Substituting $A = \varepsilon/(\eta \beta C_l)$ gives

$$m \geq \frac{\frac{N c_e}{C_l} + \frac{\varepsilon}{\eta \beta C_l} + \sqrt{\left(\frac{N c_e}{C_l} + \frac{\varepsilon}{\eta \beta C_l} \right)^2 + \frac{4 \varepsilon N}{\eta \beta C_l}}}{2}, \tag{61}$$

which concludes the proof. □

A.5 LEMMAS

Lemma 1. *Consider a von Mises-Fisher (vMF) distribution* $\mathrm{vMF}(\mu, \kappa)$*, we have*

$$\begin{aligned} D_{\mathrm{KL}}\big(\mathrm{vMF}(\mu,\kappa) \,\big\|\, \mathrm{UNIF}(\mathbb{S}^{d-1})\big) &= \kappa A_d(\kappa) + \log C_d(\kappa) + \log|\mathbb{S}^{d-1}| \\ &= \kappa A_d(\kappa) + \left(\frac{d}{2}-1\right)\log\kappa - \left(\frac{d}{2}-1\right)\log 2 - \log I_{d/2-1}(\kappa) - \log\Gamma\left(\frac{d}{2}\right), \end{aligned} \tag{62}$$

where $C_d(\kappa)$ *is the normalization constant and* $A_d(\kappa)$ *is the mean resultant length of the vMF distribution:*

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)} \qquad A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}. \tag{63}$$

As $\kappa \to 0$*, we have*

$$D_{\mathrm{KL}}\big(\mathrm{vMF}(\mu,\kappa) \,\big\|\, \mathrm{UNIF}(\mathbb{S}^{d-1})\big) = \frac{\kappa^2}{2d} + O(\kappa^4). \tag{64}$$

Proof. The density function of the von Mises-Fisher distribution on $\mathbb{S}^{d-1}$ is

$$f(z;\mu,\kappa) = C_d(\kappa)\, e^{\kappa \mu^\top z}. \tag{65}$$

The density of $\mathrm{UNIF}(\mathbb{S}^{d-1})$ is

$$p(z) = \frac{1}{|\mathbb{S}^{d-1}|}, \tag{66}$$

where

$$\log|\mathbb{S}^{d-1}| = \log 2 + \frac{d}{2}\log\pi - \log\Gamma\left(\frac{d}{2}\right). \tag{67}$$

We have

$$\begin{aligned} D_{\mathrm{KL}} &= \mathbb{E}_f\left[\log f(z) - \log p(z)\right] = \kappa A_d(\kappa) + \log C_d(\kappa) + \log|\mathbb{S}^{d-1}| \\ &= \kappa A_d(\kappa) + \left(\frac{d}{2}-1\right)\log\kappa - \left(\frac{d}{2}-1\right)\log 2 - \log I_{d/2-1}(\kappa) - \log\Gamma\left(\frac{d}{2}\right). \end{aligned} \tag{68}$$

For the modified Bessel function of the first kind $I_\nu(\kappa)$, as $\kappa \to 0$,

$$I_\nu(\kappa) = \left(\frac{\kappa}{2}\right)^\nu \sum_{m=0}^{\infty} \frac{\left(\frac{\kappa}{2}\right)^{2m}}{m!\,\Gamma(\nu+m+1)} = \frac{\left(\frac{\kappa}{2}\right)^\nu}{\Gamma(\nu+1)}\left[1 + \frac{\left(\frac{\kappa}{2}\right)^2}{\nu+1} + O(\kappa^4)\right]. \tag{69}$$

Taking $\nu = \frac{d}{2} - 1$, we have

$$I_{d/2-1}(\kappa) = \frac{\left(\frac{\kappa}{2}\right)^{d/2-1}}{\Gamma(d/2)}\left[1 + \frac{\kappa^2}{4}\cdot\frac{1}{d/2} + O(\kappa^4)\right] = \frac{\left(\frac{\kappa}{2}\right)^{d/2-1}}{\Gamma(d/2)}\left[1 + \frac{\kappa^2}{2d} + O(\kappa^4)\right]. \tag{70}$$

Taking logarithms, we have

$$\begin{aligned} \log I_{d/2-1}(\kappa) &= \left(\frac{d}{2}-1\right)\log\left(\frac{\kappa}{2}\right) - \log\Gamma\left(\frac{d}{2}\right) + \log\left[1 + \frac{\kappa^2}{2d} + O(\kappa^4)\right] \\ &= \left(\frac{d}{2}-1\right)(\log\kappa - \log 2) - \log\Gamma\left(\frac{d}{2}\right) + \frac{\kappa^2}{2d} + O(\kappa^4). \end{aligned} \tag{71}$$

Substituting $\log I_{d/2-1}(\kappa)$ into Eq. 68, we have

$$\begin{aligned} D_{\mathrm{KL}} &= \kappa A_d(\kappa) + \left(\frac{d}{2}-1\right)\log\kappa - \left(\frac{d}{2}-1\right)\log 2 \\ &\quad - \left(\frac{d}{2}-1\right)\log\kappa + \left(\frac{d}{2}-1\right)\log 2 + \log\Gamma\left(\frac{d}{2}\right) - \frac{\kappa^2}{2d} + O(\kappa^4) - \log\Gamma\left(\frac{d}{2}\right) \\ &= \kappa A_d(\kappa) - \frac{\kappa^2}{2d} + O(\kappa^4). \end{aligned} \tag{72}$$

We now focus on the term $A_d(\kappa)$. Using the small-parameter expansion,

$$I_{d/2}(\kappa) = \frac{\left(\frac{\kappa}{2}\right)^{d/2}}{\Gamma(d/2+1)}\left[1+O(\kappa^2)\right], \tag{73}$$

we have

$$\begin{aligned} A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} &= \frac{\kappa}{2}\cdot\frac{\Gamma(d/2)}{\Gamma(d/2+1)}\cdot\left[1+O(\kappa^2)\right] \\ &= \frac{\kappa}{d}+O(\kappa^3). \end{aligned} \tag{74}$$

Therefore, $\kappa A_d(\kappa) = \frac{\kappa^2}{d}+O(\kappa^4)$. Substituting it into Eq. 72, we have

$$D_{\mathrm{KL}} = \left(\frac{\kappa^2}{d}+O(\kappa^4)\right) - \frac{\kappa^2}{2d}+O(\kappa^4) = \frac{\kappa^2}{2d}+O(\kappa^4), \tag{75}$$

which concludes the proof. □

B Experimental Validation of the Assumption in Section 3

The analysis in Section 3 requires an assumption: For early data (in $\mathcal{S}_{\text{early}}$) with limited spread in all directions of the state space, the encoder is not sufficiently trained in uncovered regions ($\mathcal{S}_{\text{late}}$), so the variational posterior cannot deviate from the prior. In this section, we conduct an experiment to validate this.

In the DMC environment, we first roll out trajectories using a trained METRA policy, with the skill latent z sampled uniformly. We then split the collected trajectories $\mathcal{D}$ into two datasets, $\mathcal{D}_{y>0}$ and $\mathcal{D}_{y\leq 0}$, according to whether the average y-coordinate across all states in a trajectory is greater than 0. Finally, we train the METRA encoder on $\mathcal{D}_{y>0}$ and visualize $\|\phi(s') - \phi(s)\|_2$ for both groups.

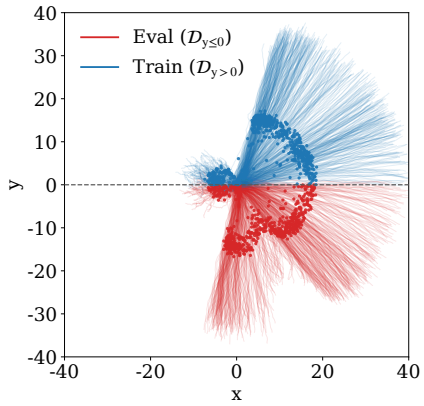


Figure 5: Visualization of sampled trajectories. Trajectories are categorized by their average y-coordinates (indicated by the dots).

We collect 1280 trajectories (620 with $y > 0$ and 660 with $y \leq 0$) and train the encoder for 200 epochs using the same training setup as METRA. The trajectories in $\mathcal{D}_{y>0}$ and $\mathcal{D}_{y\leq 0}$ are visualized in Fig. 5. This data split is designed to simulate the two state subspaces $\mathcal{S}_{\text{early}}$ and $\mathcal{S}_{\text{late}}$. Specifically, $\mathcal{D}_{y>0}$ and $\mathcal{D}_{y\leq 0}$ correspond to $\mathcal{S}_{\text{early}}$ and $\mathcal{S}_{\text{late}}$, respectively, as only $\mathcal{D}_{y>0}$ is available for the encoder here, matching the setting in Section 3, where the encoder is only sufficiently trained in the covered region $\mathcal{S}_{\text{early}}$.

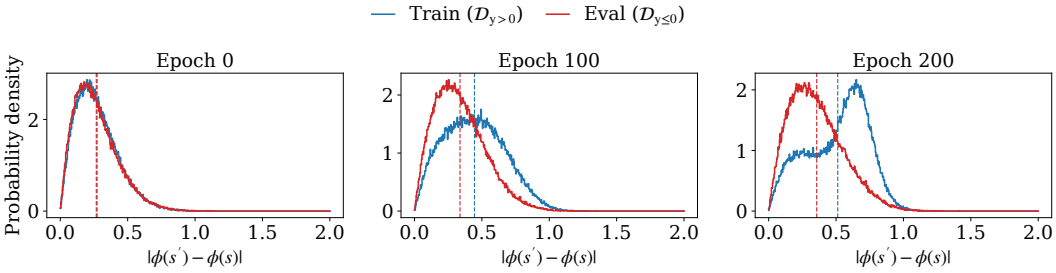


Figure 6: The histogram of $\|\phi(s') - \phi(s)\|_2$ in different training epochs.

As shown in Fig. 6, at the beginning of training, the distribution of the encoder output difference, $\|\phi(s') - \phi(s)\|_2$, is indistinguishable between the two groups. However, as training proceeds, the outputs for the $\mathcal{S}_{\text{early}}$ ($y > 0$) group change markedly, while those for the $\mathcal{S}_{\text{late}}$ group remain nearly unchanged. This validates our assumption that if the encoder is not sufficiently trained in uncovered regions, the variational posterior cannot deviate from the prior.

C ADDITIONAL EXPERIMENTS

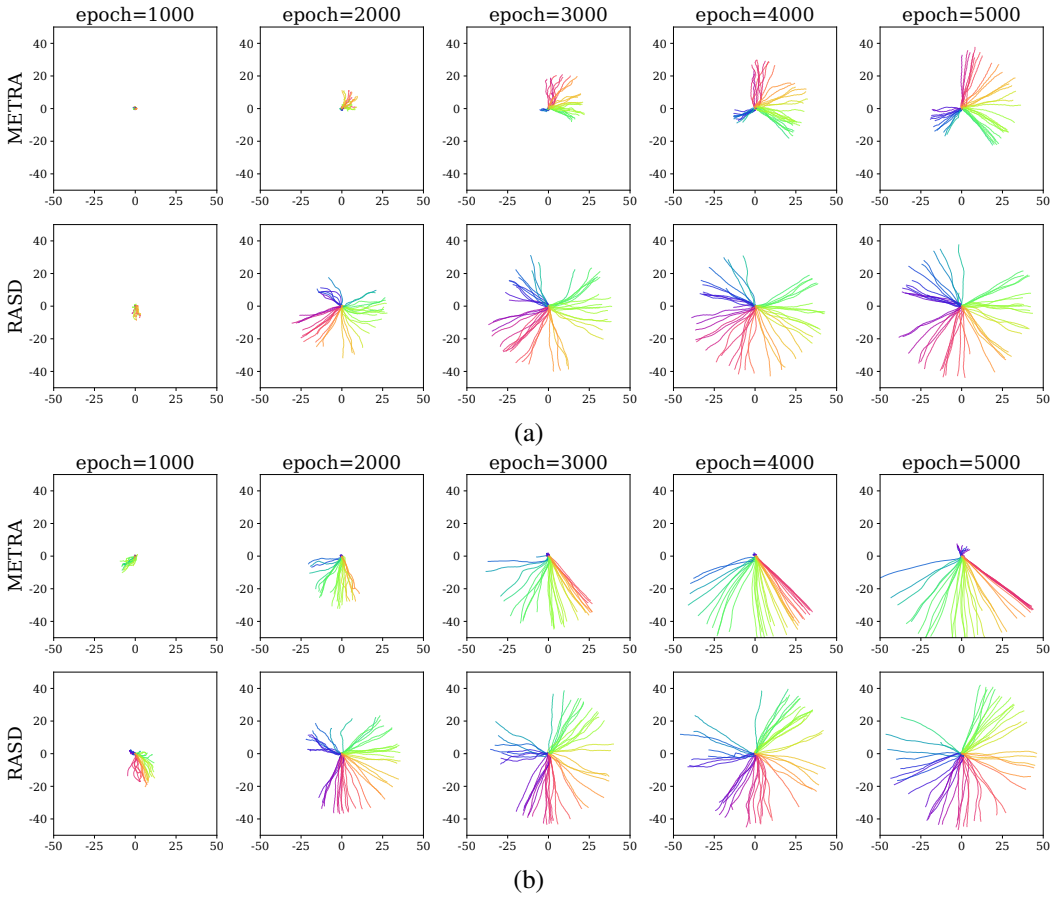


Figure 7: Visualizations of skills learned by RASD and METRA across different training epochs on the DMC Humanoid task. We plot x-y trajectories sampled from the learned policies. Different colors represent distinct skills z.

Additional Visualizations. We provide additional illustrations for the dead-zone phenomenon in Fig. 7. Each subfigure shows two trajectories, corresponding to METRA and RASD, respectively. Both trajectories of each subfigure are generated with the same random seed, thereby ensuring identical initializations of the encoders and policies.

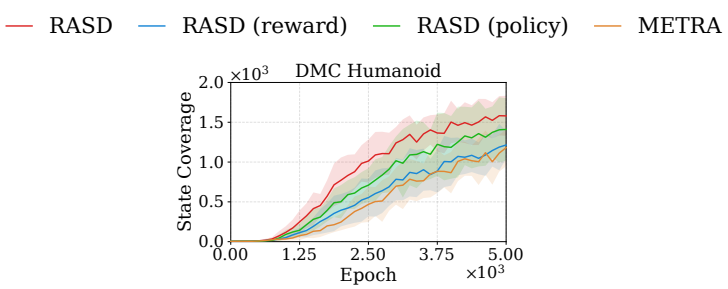


Figure 8: State coverage of RASD across all training epochs in the ablation study. RASD (reward): both buffers are used for the skill encoder, while the policy and Q-function use only the off-policy buffer. RASD (policy): both buffers are used for the policy and Q-function, while the skill encoder uses only the off-policy buffer.

Ablation on the use of recent-policy buffer in skill encoder and policy. RASD augments both intrinsic reward learning and policy training with a recent-policy buffer. As analyzed in Section 4, the recent-policy buffer enables the skill encoder to update the intrinsic reward for states at the exploration frontier in a timely manner, and enables the Q-function to promptly accumulate the effects of intrinsic rewards for these new states, jointly ensuring that the intrinsic reward encourages exploration. To validate this design, we conduct an ablation study in which the recent-policy buffer is applied to only one of the two components: either intrinsic reward learning or policy training. As shown in Fig. 8, the absence of the recent-policy buffer in either module can lead to performance degradation. Both the skill encoder and the policy should be augmented with the recent-policy buffer to maximize exploration. This justifies the necessity of applying the recent-policy buffer to both modules.

D EXPERIMENTAL DETAILS

D.1 SETUPS

Environments. We evaluate RASD on five complex robotic locomotion environments, including Ant and Humanoid from OpenAI Gym (Todorov et al., 2012; Brockman et al., 2016), and Hopper, Quadruped, and DMC Humanoid from DMControl (Tassa et al., 2018). In all environments, we use the proprioceptive state of the robot as the observation. Following Park et al. (2024), we ensure the xy location (or x location) is included in the robot observation.

Metrics. We employ two main metrics for evaluation:

- **State coverage** measures the agent's ability to explore the state space. Following Park et al. (2024), this metric computes state coverage by counting the unique 1×1 x-y bins visited by the agent (or 1-unit x-axis bins for Hopper).
- **Downstream task performance** evaluates the utility of learned skills in downstream tasks. We consider both a zero-shot setting and a task-specific hierarchical control setting. To assess zero-shot performance, we roll out the downstream task environment with skills derived from randomly sampled goals and report the success rate over all sampled goals. To assess hierarchical control performance, we use the learned skills as a low-level controller and train an additional high-level controller to optimize performance on the downstream task. The downstream task performance is then reported as the hierarchical control performance of the pretrained skills. Further details on the downstream tasks can be found in Appendix D.2, while additional information about the high-level controller is provided in Appendix D.3.

D.2 DOWNSTREAM TASK

Zero-shot performance. We evaluate the zero-shot performance of the learned skills on a goal-reaching task in Hopper and DMC Humanoid. For RASD, METRA, and LSD, the policy is driven by the skill $z \propto \phi(o_g) - \phi(o)$, where ϕ is the skill encoder, o is the current observation, and o_g is the current observation with its location components replaced by the goal location g. For DIAYN, the skill latent is selected based on the output of the skill discriminator at the goal state. Note that since CIC and DADS learn $q(s|z)$ rather than $q(z|s)$, we cannot select the skill latent z using the goal state. Therefore, we do not include CIC and DADS in this evaluation. The skill is recomputed from the current observation at every step. The goal location is sampled uniformly from the range $[-10, 10]^2$ for DMC Humanoid, and $[-10, 10]$ for Hopper. Success is defined geometrically, and an episode counts as a success if the agent ever comes within a distance of less than 3 of the goal. We report the fraction of the 50 sampled goals that are reached.

Hierarchical control performance. We evaluate the hierarchical control performance of the learned skills in Hopper and DMC Humanoid on a goal-reaching task, where the agent must move its torso to a goal location. The agent will receive a reward of 1.0 if it is sufficiently close to the goal (i.e., within a distance of less than 1.5), and 0.0 otherwise. The goals are randomly sampled from the range $[-5, 5]^2$ for DMC Humanoid and $[-10, 10]$ for Hopper, which remain fixed for the duration of an episode. The pretrained skills are frozen throughout the training process of the high-level controller.

D.3 IMPLEMENTATION DETAILS

For the baselines, we adopt the implementation of METRA (Park et al., 2024), DIAYN (Eysenbach et al., 2019), CIC (Laskin et al., 2022), DADS (Sharma et al., 2020), and LSD (Park et al., 2022) from the codebase provided in Zheng et al. (2025). Code to implement RASD is provided in the supplementary material.

RASD shares the same hyperparameters as the baselines, which are consistent with METRA. We list these hyperparameters in Table 3.

Table 3: Common hyperparameters for unsupervised skill discovery methods.

Hyperparameter	Value
Encoder for state-based tasks	MLP
# hidden layers	2
# hidden units per layer	1024
Learning rate	0.0001
Optimizer	Adam (Kingma & Ba, 2015)
Minibatch size	256
Target network smoothing coefficient	0.995
Entropy coefficient	auto-adjust (Haarnoja et al., 2018)
Total horizon length	200 (Ant, Hopper, Quadruped, DMC Humanoid), 1000 (Humanoid)
# epochs	5000 (Ant, Hopper, Quadruped, DMC Humanoid), 50000 (Humanoid)
# episodes per epoch	8
# gradient steps per epoch	200 (Hopper, Quadruped, DMC Humanoid), 50 (Ant, Humanoid)
Skill latent dimension d	2 (Ant, Hopper, Humanoid, DMC Humanoid), 4 (Quadruped), 64 (All environments in CIC)
Replay buffer size N_{off}	10^6
Discount factor γ	0.99
METRA ϵ	10^{-3}
METRA initial λ	30

The additional hyperparameters of RASD are listed in Table 4. Both buffers are written to with every collected transition, and each minibatch is split so that a fraction p of it is drawn from the recent-policy buffer with uniform sampling, while the remaining fraction is drawn from the off-policy buffer. The off-policy buffer size is the same as the replay buffer size listed in Table 3.

Table 4: Additional hyperparameters for RASD.

Hyperparameter	Value
Recent-policy-buffer sampling ratio p	0.75
Recent-policy-buffer capacity N_{on}	10^5

For hierarchical control tasks, we use a SAC (Haarnoja et al., 2018) agent as the high-level controller. The trained skills serve as the low-level controller, and their parameters are fixed during the training of the hierarchical control agent. The hyperparameters are shown in Table 5.

Table 5: Hyperparameters for high-level controllers.

Hyperparameter	Value	Hyperparameter	Value
Learning rate	0.0001	Option timesteps length	25
Total horizon length	200	Replay buffer batch size	256
# hidden layers	2	# hidden units per layer	1024
Temperature α	0.1 (auto-adjust)	# epochs	10000
# episodes per epoch	8		